View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Electronic Archive of Kyiv Polytechnic Institute

эффективно управлять объектами безопасности. Компонентная технология модели позволяет интегрировать сервисы управления объектами безопасности в исходный код CSP с использованием драйверов (kernel-mode drivers) блокировки страниц виртуальной памяти для платформ Microsoft® Win32®.

Разработка применяется в пользовательских системах защиты информации и системах защиты сетевого трафика для обеспечения конфиденциальности и целостности передаваемой и хранимой информации, а также в пользовательских системах, обрабатывающих и управляющих критической информацией, с целью защиты информации от возможных каналов утечки, связанных с особенностью управления памятью ОС Microsoft® Win32®. Однако разработанная технология не учитывает другие каналы возможной утечки информации, поэтому используется с другими системными компонентами.

Литература: 1. Microsoft Developer Network Library – January 2000. Platform SDK (Base Services). Microsoft Corporation. 2. Windows 2000 Driver Development Kit. Kernel-mode drivers. Microsoft Corporation. 3. Windows 98 Driver Development Kit. Virtual device drivers. Microsoft Corporation.

УДК 681.142.35

АЛГОРИТМЫ ШИФРОВАНИЯ ДЛЯ ПЕРЕДАЧИ ДАННЫХ В ОТКРЫТЫХ СЕТЯХ

Сергей Макаренко, Андрей Брусникин
Национальный технический университет Украины "КПИ"

Анотація: **Надано стислий перелік вимог до алгоритмів кодування, що використовуються у відкритих мережах.**

Summary: **The list of requirements is given below for the security algorithms implemented for use in open network.**

Ключевые слова: **Шифр, RSA, DES, IDEA, ГОСТ 28147-89.**

I Введение

Бурное развитие электронной коммерции и все возрастающее число пользователей заставляют серьезно задуматься о проблемах безопасности открытых каналов передачи данных. Отсутствие руководящих органов привело к тому, что каждый член сети должен заботиться о своей безопасности сам.

Подключение к открытым (глобальным) сетям, таким как Интернет, существенно увеличивает эффективность работы и открывает множество новых возможностей. В то же время необходимо позаботится о создании системы защиты информационных ресурсов от желающих их использовать, модифицировать или просто уничтожить. Защита информации подразумевает поддержание целостности, доступности и, если необходимо, конфиденциальности информации и ресурсов, используемых для ввода, хранения, обработки и передачи данных. Для решения комплексной проблемы защиты необходимо сочетание законодательных, организационных и программно - технических мер [1].

Данная работа посвящена влиянию специфики открытых сетей на решение задач обеспечения безопасности передачи данных и создания библиотек на основе лучших алгоритмов. Показано, как особенности данных отразились на развитии современных алгоритмов их шифрования.

II Общие положения [1–5]

1. Абсолютно стойкие алгоритмы уже давно существуют, но до сих пор не нашли широкого применения в открытых сетях.

Первым абсолютно стойким алгоритмом был алгоритм Вернама, а его теоретическое обоснование разработано Шенноном.

Максимально возможная неопределенность блока данных фиксированного размера достигается, когда все возможные значения этого блока равновероятны. В этом случае она равна размеру блока в битах. Таким образом, неопределенность ключа K не превышает его длины:

$$H(K) \le K.$$

Условие абсолютной стойкости для шифров, удовлетворяющих принципу Кирхгофа,

$$|K| \ge H(K) = H(E) \ge H(T) = |T|,$$

где E – шифр, а T – исходные данные.

Для того чтобы шифр, построенный по принципу Кирхгофа, был абсолютно стойким, необходимо, чтобы

размер использованного для шифрования ключа был не меньше размера шифруемых данных, т. е.

$$|K| \geq |T|.$$

Точное равенство возможно только в том случае, если все возможные значения ключа равновероятны. Это эквивалентно условию, что биты ключа равновероятны и статистически независимы друг от друга.

Примером абсолютно стойкого шифра может служить одноразовая гамма Вернама - наложение с помощью некоторой бинарной операции ° на открытые данные T ключа K такого же размера, составленного из статистически независимых битов, которые принимают возможные значения с одинаковой вероятностью

$$T' = T \circ K.$$

Используемая для наложения гаммы операция должна удовлетворять следующим условиям: уравнение шифрования должно быть однозначно разрешимо относительно

открытых данных при известных зашифрованных и ключе;

ключа при известных открытых и зашифрованных данных.

По указанной причине обычно выбирают наиболее удобную в реализации операцию - побитовое суммирование по модулю 2 (или побитовое исключающее ИЛИ), так как она: 1) требует для своей реализации простейшей логики из всех возможных операций; 2) обратная самой себе, поэтому для шифрования и дешифрования применяется одна и та же процедура.

Другим, абсолютно стойким алгоритмом является семейство блокнотных алгоритмов. На предварительном этапе происходит обмен “блокнотами”, представляющими избыточный набор символов, которые могут быть использованы при передаче сообщения. Примером такого блокнота может быть книга или беспорядочный кусок информации (предпочтительнее). На этапе шифрования происходит сопоставление каждого символа шифруемого текста различным позициям этого символа в блокноте. Следовательно, достигается абсолютная неопределенность передаваемой информации.

Таким образом, абсолютно стойкие шифры требуют использования ключа, по размеру не меньшего шифруемых данных. Этот ключ должен быть и у отправителя, и у получателя, то есть его необходимо предварительно доставить им, а для этого необходим защищенный канал. Наряду с потенциально незащищенным каналом для передачи зашифрованных данных необходимо существование защищенного канала для передачи такого же по размеру ключа. Это не всегда приемлемо по экономическим соображениям, поэтому подобные системы применяются лишь в исключительных случаях для защиты сведений, представляющих особую ценность. В подавляющем большинстве реальных систем шифрованной связи используются алгоритмы, не обладающие абсолютной стойкостью и поэтому называемые несовершенными шифрами.

2. Недоступность алгоритма не повышает безопасность шифра, за стандарт принимаются открытые алгоритмы.

Не существует способа получить точное значение трудоемкости криптоанализа. Все оценки базируются на проверках устойчивости шифров к известным на текущий момент видам криптоанализа, и нет гарантии, что в ближайшем будущем не будут разработаны новые методы анализа, существенно снижающие трудоемкость. Сказанное выше означает, что при текущем положении дел в криптографии стойкость абсолютно всех шифров, за исключением совершенных, не может быть доказательно обоснована. Вместо этого она обоснована эмпирически как устойчивость к известным сегодня видам криптоанализа, но никто не может дать гарантии, что завтра не будет изобретен вид криптоанализа, успешный именно для данного конкретного шифра. Вот почему не стоит доверять "новейшим шифрам" - они не прошли проверку временем. По этой же самой причине не разумно доверять криптоалгоритмам, которые их авторы держат в секрете - даже при отсутствии злонамеренно оставленных там "люков" нет совершенно никакой гарантии того, что алгоритм был исследован со всей необходимой тщательностью. Примером шифра, предложенного для внедрения без разглашения его алгоритма, является Clipper от АНБ США. Требование перенастраиваемости алгоритма появляется вследствие того, что в распоряжении злоумышленника рано или поздно может оказаться описание алгоритма, его программная или аппаратная реализация. Для того, чтобы в этом случае не пришлось заменять алгоритм полностью, во всех узлах шифрования, где он используется, он должен содержать легко сменяемую часть. Это приводит к принципу Кирхгофа: шифр определяется как параметризованный алгоритм, состоящий из процедурной части, то есть описания того, какие именно операции и в какой последовательности выполняются над шифруемыми данными, и параметров - различных элементов данных, используемых в преобразованиях. Раскрытие только процедурной части не должно приводить к увеличению вероятности успешного дешифрования сообщения злоумышленником выше допустимого предела. По этой причине, а также в силу того, что рассекречивание этой части достаточно вероятно само по себе, нет особого смысла хранить ее в секрете. В секрете держится некоторая часть параметров алгоритма, которая называется ключом шифра:

$$T' = E(T) = E_K(T),$$

где К - ключ шифра.

Использование принципа Кирхгофа позволяет сделать некоторые выводы о построении шифров. Разглашение конкретного шифра (алгоритма и ключа) не приводит к необходимости полной замены реализации всего алгоритма, достаточно заменить только раскрытый ключ. Ключи можно отчуждать от остальных компонентов системы шифрования (хранить отдельно от реализации алгоритма, в более надежном месте) и загружать их только по мере необходимости и на время выполнения шифрования (это значительно повышает надежность системы в целом). Появляется возможность для точной оценки "степени неопределенности" алгоритма шифрования - она просто равна неопределенности используемого ключа:

$$H(E_K) = H(K).$$

Соответственно, становится возможным оценить вероятность и трудоемкость успешного дешифрования, то есть количество вычислительной работы, которую необходимо выполнить для этого.

3. Подавляющее большинство процессоров имеет 32 – разрядную архитектуру, что накладывает свои ограничения на программную реализацию шифра. Для сравнения приведем характеристики русского ГОСТ 28147-89 (далее ГОСТ) и американского (DES) стандартов, разработанных приблизительно в одно и то же время (табл. 1).

Таблица 1

П А Р А М Е Т Р	ГОСТ	DES
1. Размер блока шифрования	64 бита	64 бита
2. Длина ключа	256 бит	56 бит
3. Число раундов	32	16
4. Узлы замен (S-блоки)	не фиксированы	фиксированы
5. Длина ключа для одного раунда	32 бита	48 бит
6. Схема выработки раундового ключа	Простая	сложная
7. Начальная и конечная перестановки битов	Нет	есть

Так как DES разрабатывался во времена 8-разрядной архитектуры, то сегодня возникает ряд неудобств при программной реализации этого алгоритма на современных ПК, что приводит к тому, что ГОСТ, несмотря на большую длину ключа и большее число раундов, работает быстрее.

Специально созданным для 32-битных машин, а также существенно более быстрым, чем DES алгоритмом, является популярный Blowfish.

Предстоящий переход на 64-разрядную архитектуру порождает требование к будущим алгоритмам учитывать особенности этой архитектуры, хотя это и не обязательно для аппаратной реализации.

4. Иногда не представляется возможным использовать защищенный канал на предварительном этапе. Это привело к разделению алгоритмов на классические – симметричные и новые, асимметричные алгоритмы шифрования.

В "симметричных" криптоалгоритмах (DES, ГОСТ, Blowfish, RC5, IDEA) один и тот же ключ используется как для шифрования, так и для восстановления открытого сообщения. Поэтому этот ключ является секретным. Достоинством этих алгоритмов является их хорошая теоретическая изученность, в том числе обоснование криптостойкости. По сравнению с "асимметричными" алгоритмами следует отметить относительную простоту как программной, так и аппаратной реализации, более высокую скорость работы в прямом и в обратном направлении, а также обеспечение необходимого уровня защиты при использовании существенно более коротких ключей. К основным недостаткам следует отнести необходимость обеспечения дополнительных мер секретности при распространении ключей, связанные с этим проблемы и, возможно, затраты, а также тот факт, что алгоритмы с секретным ключом работают только в условиях полного доверия корреспондентов друг другу, т.к. не позволяют реализовать настоящую "цифровую подпись".

В "асимметричных" методах (RSA, ECC) прямое и обратное криптопреобразования выполняются с использованием различных полу-ключей, которые не имеют между собой легко прослеживаемых связей, позволяющих по одному полу-ключу вычислить другой. Поэтому один из полу-ключей открыто публикуется для того, чтобы каждый мог зашифровать сообщение или проверить цифровую подпись. Расшифровать такое сообщение или поставить подпись может только тот, кто знает второй - секретный - полу-ключ. Такие алгоритмы, по сравнению с "симметричными", более требовательны к вычислительным ресурсам и, следовательно, их реализация и использование обходится дороже. На сегодняшний день криптостойкость "асимметричных" алгоритмов обоснована хуже, чем криптостойкость "симметричных" алгоритмов. Но они работают там, где "классические" криптосхемы неприменимы: позволяют реализовать различные хитроумные протоколы типа цифровой подписи, открытого распределения ключей и надежной аутентификации в сети, устойчивой даже к полному перехвату трафика. Однако применение асимметричных

методов привело к появлению нового ряда проблем. Главной из них является проблема получения достоверного открытого ключа адресата.

III Основные алгоритмы шифрования [1, 3, 6–8]

Функция шифрования

Процесс шифрования состоит из набора раундов-шагов, на каждом шаге выполняются следующие действия. Входной блок делится пополам на старшую (L) и младшую (R) части. Вычисляется значение функции шифрования от младшей части (R) и раундового ключа (k) X=f(R,k). Используемая на данном шаге функция и называется функцией шифрования раунда. Она может быть одна для всех раундов или индивидуальна для каждого раунда. В последнем случае функции шифрования различных раундов одного шифра отличаются, как правило, лишь в деталях. Формируется выходной блок, его старшая часть равна младшей части входного блока L'=R, а младшая часть – это результат выполнения операции побитового исключающего ИЛИ (обозначим его (+)) для старшей части входного блока и результата вычисления функции шифрования R'=L(+)f(R,k).

Алгоритм DES

DES (Data Encryption Standart) – это симметричный алгоритм шифрования, т. е. один ключ используется как для зашифровки, так и для расшифровки сообщений. Разработан фирмой IBM и утвержден правительством США в 1977 как официальный стандарт. DES имеет блоки по 64 бит и основан на 16-ти кратной перестановке данных, также для зашифровки использует ключ в 56 бит. Существует несколько режимов DES, например Electronic Code Book (ECB) и Cipher Block Chaining (CBC). 56 бит - это 8 семибитовых ASCII символов, т. е. пароль не может быть длиннее 8 букв. Если вдобавок использовать только буквы и цифры, то количество возможных вариантов будет гораздо меньше максимально возможных 2^{56}.

Один из шагов алгоритма DES

Входной блок данных делится пополам на левую (L') и правую (R') части. После этого формируется выходной массив так, что его левая часть L" представлена правой частью R' входного, из 32-битового слова R' с помощью битовых перестановок формируется 48-битовое слово. К полученному 48-битовому слову и 48-битовому раундовому ключу применяется операция XOR . Результирующее 48-битовое слово разбивается на 8 6-битовых групп, каждая 6-битовая группа с помощью соответствующего S-box'a заменяется на 4-битовую группу и из полученных восьми 4-битовых групп составляется 32-битовое слово. К полученному слову и L' применяется XOR, в результате получается R". Можно убедиться, что все проведенные операции могут быть обращены, и расшифровка может осуществляться за число операций, линейно зависящее от размера блока. После нескольких таких проходов можно считать, что каждый бит выходного блока шифровки может зависеть от каждого бита сообщения.

Алгоритм “тройной DES”

Так как текст, зашифрованный двойным DES (встреча на середине (meet in the middle)), оказывается хрупким при криптографической атаке, то текст шифруется 3 раза DES. Таким образом, длина ключа возрастает до 168-бит (56x3). Не всегда применение тройного DES означает увеличение уровня безопасности сообщения. Типы тройного шифрования DES:

– DES-EEE3: шифруется 3 раза с 3 различными ключами.
– DES-EDE3: 3 DES операции шифрование - дешифрование - шифрование с 3 различными ключами.
– DES-EEE2 и DES-EDE2: как и предыдущие, за исключением того, что первая и третья операции используют одинаковый ключ.

В табл. 2 приведено сравнение различных видов DES шифрования.

Таблица 2

# Шифрования	# Ключей	Вычисление (Computation)	Хранение (Storage)	Тип атаки
Одиночный	1	2^{56}	-	known plaintext
Одиночный	1	2^{38}	2^{38}	chosen plaintext
Одиночный	1	-	2^{56}	chosen plaintext
Двойной	2	2^{112}	-	known plaintext
Двойной	2	2^{56}	2^{56}	known plaintext
Двойной	2	-	2^{112}	chosen plaintext
Тройной	2	2^{112}	-	known plaintext
Тройной	2	2^{56}	2^{56}	2^{56} chosen plaintext
Тройной	2	$2^{(120-t)}$	-	2^{t} known plaintext
Тройной	2	-	2^{56}	chosen plaintext

ISBN 966-622-022-9
Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні, вип. 2, 2001 р.

Окончание таблицы 2

# Шифрования	# Ключей	Вычисление (Computation)	Хранение (Storage)	Тип атаки
Тройной	3	2^{112}	2^{56}	known plaintext
Тройной	3	2^{56}	2^{112}	chosen plaintext

Алгоритм ГОСТ

ГОСТ предусматривает 3 режима шифрования (простая замена, гаммирование, гаммирование с обратной связью) и один режим выработки имитовставки. Первый из режимов шифрования предназначен для шифрования ключевой информации и не может использоваться для шифрования других данных, для этого предусмотрены два других режима шифрования. Режим выработки имитовставки (криптографической контрольной комбинации) предназначен для имитозащиты шифруемых данных, то есть для их защиты от случайных или преднамеренных несанкционированных изменений.

Алгоритм построен по тому же принципу, что и DES – это классический блочный шифр с секретным ключом – однако отличается от DES'а большей длиной ключа, большим количеством раундов, и более простой схемой построения самих раундов.

Из-за намного большей длины ключа ГОСТ гораздо устойчивей DES'а к вскрытию "грубой силой" – путем полного перебора по множеству возможных значений ключа.

Функция шифрования ГОСТа гораздо проще функции шифрования DES'а, она не содержит операций битовых перестановок, коими изобилует DES и которые крайне неэффективно реализуются на современных универсальных процессорах (хотя очень просто аппаратно - путем разводки проводников в кристалле или на плате). В силу сказанного, при вдвое большем количестве раундов (32 против 16) программная реализация ГОСТа на процессорах Intel x86 более чем в 2 раза превосходит по быстродействию реализацию DES'а. Естественно, сравнивались близкие к оптимуму по быстродействию реализации.

Из других отличий ГОСТа от DES'а надо отметить следующие:

На каждом раунде шифрования используется "раундовый ключ", в DES'е он 48-битовый и вырабатывается по относительно сложному алгоритму, включающему битовые перестановки и замены по таблице, в ГОСТе он берется как фрагмент ключа шифрования. Длина ключа шифрования в ГОСТе равна 256 битам, длина раундового ключа - 32 битам, итого получаем, что ключ шифрования ГОСТа содержит 256/32=8 раундовых ключей. В ГОСТе 32 раунда, следовательно, каждый раундовый ключ используется 4 раза, порядок использования раундовых ключей установлен в ГОСТе и различен для различных режимов.

Таблица замен в ГОСТе - аналог S-блоков DES'а - представляет собой таблицу (матрицу) размером 8x16, содержащую число от 0 до 15. В каждой строке каждое из 16-ти чисел должно встретиться ровно 1 раз. В отличие от DES'а, таблица замен в ГОСТе одна и та же для всех раундов и не зафиксирована в стандарте, а является сменяемым секретным ключевым элементом. От качества этой таблицы зависит качество шифра. При "сильной" таблице замен стойкость шифра не опускается ниже некоторого допустимого предела даже в случае ее разглашения. И наоборот, использование "слабой" таблицы может уменьшить стойкость шифра до недопустимо низкого предела. Никакой информации по качеству таблицы замен в открытой печати России не публиковалось, однако существование "слабых" таблиц не вызывает сомнения - примером может служить "тривиальная" таблица замен, по которой каждое значение заменяется самим собой. Это делает ненужным для компетентных органов России ограничивать длину ключа – можно просто поставить недостаточно "сильную" таблицу замен.

В ГОСТе, в отличие от DES'а, нет начальной и конечной битовых перестановок шифруемого блока, которые, по мнению ряда специалистов, не влияют существенно на стойкость шифра, хотя влияют (в сторону уменьшения) на эффективность его реализации.

Шифр Blowfish

Blowfish – это 64-битный блочный шифр разработанный Шнайером (Schneier) в 1993 году. Это шифр Файстела (Feistel), и каждый проход состоит из зависимой от ключа перестановки и зависимой от ключа с данными замены. Все операции основаны на операциях XOR и прибавлениях к 32-битным словам (XORs and additions on 32-bit words). Ключ имеет переменную длину (максимально 448 бит) и используется для генерации нескольких подключевых массивов (subkey arrays). Шифр был создан специально для 32-битных машин и существенно быстрее DES.

Шифр RC5

RC5 – это довольно быстрый блочный шифр разработанный Ривестом для RSA Data Security. Этот алгоритм параметричный, т.е. с переменным размером блока, длиной ключа и переменным числом проходов. Размер блока может быть 32, 64, или 128 битов. Количество проходов в промежутке от 0 до 2048 бит. Параметричность такого рода дает гибкость и эффективность шифрования.

RC5 состоит из ввода ключа (key expansion), шифрования и дешифровки. При вводе ключа вводятся

также количество проходов, размер блока и т.д. Шифрование состоит из 3 примитивных операций: сложения, побитового XOR и чередования (rotation). Исключительная простота RC5 делает его простым в использовании. RC5 текст, также как и RSA, может быть дописан в конец письма в зашифрованном виде.

Безопасность RC5 основывается на зависящих от данных чередования и смешивания результатах различных операций. RC5 с размером блока 64 бита и 12 или более проходов обеспечивает хорошую стойкость к дифференциальному и линейному криптоанализу.

Шифр IDEA

IDEA (International Data Encryption Algorithm) - это вторая версия блочного шифра, разработанная К. Лейем (Lai) и Д. Месси (Massey) в конце 80-х. Это шифр, состоящий из 64-битных повторяющихся блоков со 128-битным ключом и восемью проходами (rounds). Хотя этот шифр не является шифром Файстела, дешифровка выполняется по тому же принципу, что и шифрование. Структура шифра была разработана для легкого воплощения как программно, так и аппаратно, и безопасность IDEA основывается на использовании трех не совместимых типов арифметических операций над 16-битными словами. Скорость программного IDEA сравнима со скоростью DES.

Один из принципов создания IDEA – затруднить дифференциальный криптоанализ. Ни одна линейная криптоаналитическая атака не закончилась успешно, как и не было выявлено алгебраически слабых мест. Самый полный анализ провел Daemen. Он открыл большой класс 2^{51} слабых ключей, при использовании которых в процессе шифрования ключ может быть обнаружен и восстановлен. Однако, в IDEA существует 2^{128} возможных вариантов ключей, поэтому это открытие не влияет на практическую безопасность шифра.

Шифр RSA

RSA (авторами являются Rivest, Shamir и Alderman) – это система с открытым ключом (public-key), предназначенная как для шифрования, так и для аутентификации. Была разработана в 1977 году. Она основана на трудности разложения очень больших целых чисел на простые сомножители.

RSA очень медленный алгоритм. Для сравнения: на программном уровне DES по меньшей мере в 100 раз быстрее RSA, на аппаратном – в 1000-10000 раз, в зависимости от выполнения.

Алгоритм RSA состоит в следующем:

– Для двух очень больших целых чисел P и Q определяются N=PQ и M=(P - 1)(Q - 1).

– Выбирается случайное целое число D, взаимно простое с M, и вычисляется E = (1 mod M)/D.

– D и N публикуются как открытый ключ, а E сохраняется в тайне.

– Пусть S – сообщение. Его длина определяется значением выражаемого им целого числа и находится в интервале (1, N). S превращается в шифровку возведением в степень D по модулю N и отправляется получателю $S' = (S^D \bmod N)$.

– Получатель сообщения расшифровывает его, возведя в степень E (число E ему уже известно) по модулю N, т. к. $S = ((S')^D \bmod N) = (S^{DE} \bmod N)$.

Пакет PGP

Pretty Good Privacy(PGP) - это программный пакет разработанный Филипом Циммерманом (Philip Zimmerman), который обеспечивает шифровку почты и файлов. Циммерман взял существующие криптосистемы и криптографические протоколы и разработал бесплатную (freeware) программу для различных платформ. Она обеспечивает шифрование сообщений, цифровые подписи и совместимую почту (email compatibility).

Алгоритмы, используемые для шифрования сообщений - это RSA для передачи ключа и IDEA для самого шифрования сообщений. Цифровые подписи достигаются при использовании RSA для подписи и MD5 для вычисления дайджеста сообщения (message digest). PGP использует ZIP компрессию, а также маскирует координаты и данные отправителя, что немного осложняет процесс анализа трафика. Совместимость почты достигается путем использования Radix-64 конвертации (conversion).

PGP проверена огромным количеством людей, ее исходные тексты опубликованы в Интернете, но поскольку в ней используется RSA, то степень защиты сообщения зависит от длины ключа (чем больше, тем лучше).

Пакет MIT PGP версии 2.6 и позже – это вполне легальная бесплатная программа для не коммерческого использования. Пакет Viacrypt версии 2.7 и позже – это коммерческий продукт.

IV Выводы

На сегодняшний день наилучшую степень защиты данных для домашних пользователей при работе в открытых сетях обеспечивает пакет PGP. Для защиты данных на уровне организаций целесообразно использовать коммерческие версии пакетов шифрования, основанных на асимметричных алгоритмах (пакеты на основе алгоритмов RSA, ECC). Максимальная защищенность может быть получена при выборе оптимального алгоритма под конкретную ситуацию.

Литература: 1. Nichols R. K. ICSA. Guide to Cryptography. McGrow-Hill, 1999, Part 2, 4. 2. Вербіцький О. В. Вступ до криптографії. Видавництво Науково-Технічної літератури. Львів, 1998. 3. Винокуров А. Проблема аутентификации данных и блочные шифры (эл. книга: http://maul.samara.net/~kodan/yandex/yandex.html). 4. Водолазский В. Коммерческие системы шифрования: основные алгоритмы и их реализация. Часть 1. // Монитор. - 1998. - № 6-7. - с. 14-19. 5. Ковалевский В., Максимов В. Криптографические методы. // КомпьютерПресс. - 1993. - № 5. - с. 31-34. 6. Мафтик С. Механизмы защиты в сетях ЭВМ. - М.: Мир, 1996. 7. Zimmermann P. A Proposed Standard Format for RSA Cryptosystems // Advances in Computer Security, v. III, 1988, edited by Rein Turn, Artech House. 8. Garfinkel S. Pretty Good Privacy // O'Reilly & Associates, 1995.